Protein Engineering, Design & Selection vol. 24 no. 5 pp. 455–461, 2011
Published online January 31, 2011 doi:10.1093/protein/gzr003

Can computationally designed protein sequences improve secondary structure prediction?

Rajkumar Bondugula[1,4], Anders Wallqvist[1] and Michael S. Lee[1,2,3,5]

[1]Biotechnology HPC Software Applications Institute, Telemedicine and Advanced Technology Research Center, U.S. Army Medical Research and Materiel Command, Fort Detrick, MD 21702, USA, [2]Computational Sciences and Engineering Branch, US Army Research Laboratory, Aberdeen Proving Ground, MD 21005, USA and, [3]Department of Cell Biology and Biochemistry, US Army Medical Research Institute of Infectious Diseases, Fort Detrick, MD 21702, USA and [4]Present address: Machine Learning Group, Online Business Unit of Sears Holdings Corporation, Hoffman Estates, IL 60192, USA

[5]To whom correspondence should be addressed.
E-mail: michael.lee@amedd.army.mil

Received July 29, 2010; revised November 30, 2010; accepted January 5, 2011

Edited by Michael Sternberg, Board Member for PEDS

Computational sequence design methods are used to engineer proteins with desired properties such as increased thermal stability and novel function. In addition, these algorithms can be used to identify an envelope of sequences that may be compatible with a particular protein fold topology. In this regard, we hypothesized that sequence-property prediction, specifically secondary structure, could be significantly enhanced by using a large database of computationally designed sequences. We performed a large-scale test of this hypothesis with 6511 diverse protein domains and 50 designed sequences per domain. After analysis of the inherent accuracy of the designed sequences database, we realized that it was necessary to put constraints on what fraction of the native sequence should be allowed to change. With mutational constraints, accuracy was improved vs. no constraints, but the diversity of designed sequences, and hence effective size of the database, was moderately reduced. Overall, the best three-state prediction accuracy (Q_3) that we achieved was nearly a percentage point improved over using a natural sequence database alone, well below the theoretical possibility for improvement of 8–10 percentage points. Furthermore, our nascent method was used to augment the state-of-the-art PSIPRED program by a percentage point.

Keywords: computational protein design/fuzzy nearest neighbor/RosettaDesign/secondary structure prediction

Introduction

Computational sequence design (CSD) is becoming increasingly useful in engineering proteins for increased thermal stability, novel function and new folds. Beyond its utility in the protein engineering community, CSD has been used to study the fundamental question of sequence–structure compatibility (Larson *et al*., 2002) and has been evaluated in the realm of protein structure prediction (Larson *et al*., 2003; am Busch *et al*., 2009; Schmidt Am Busch *et al*.), including enhancing homology detection (Larson *et al*., 2003), fold recognition (am Busch *et al*., 2009; Schmidt Am Busch *et al*.), *ab initio* model detection (Koehl and Levitt, 2002) and active site residue identification (Pei *et al*., 2003; Cheng *et al*., 2005). For example, Levitt and co-workers conjectured that *ab initio* models that yield designed sequences closest to the query sequence must also be closest to the native structure (Koehl and Levitt, 2002). am Busch *et al*. (2009) and Pei *et al*. (2003) found that computationally designed sequences improved the position-specific scoring matrices used to detect other proteins with the same fold. Pei *et al*. (2003) and Cheng *et al*. (2005) used computationally designed sequences to improve the detection of active site residues by comparing natural sequence substitution rates with mutational rates from computational design. The theory behind their idea is that computational design is often singularly focused on thermal stability. Therefore, if it calls for mutations to naturally conserved residues, those residues may be conserved to preserve a protein function.

More fundamental than detection of homologous proteins for structure prediction, secondary structure prediction (SSP) is a relatively mature problem starting with the work of Chou and Fasman (1974). Traditionally, the goal is to assign each residue in a query sequence one of the following eight secondary structure states: α-helix (H), 3_{10}-helix (G), π-helix (I), isolated β-bridge (B), β-strand (E), bend (S), turn (T) and coil (C). For simplicity, the eight secondary structure states can be grouped into three categories: [H, G, I]→helix, [E, B]→strand and [C, T, S]→coil. SSP benefits from the ever-increasing number of known protein structures and sequences. Protein structures provide references to how different stretches of sequences translate to secondary structure. In addition, protein sequences from the ever-increasing database of sequenced genomes add to the profile of a query sequence, thereby improving the search (using e.g. PSI-BLAST) for similar sequence fragments in the Protein Data Bank (PDB) (Berman *et al*., 2002). SSP accuracy has been increasing (Fig. 1a) in tandem with the increasing number of known unique PDB sequences (Fig. 1b). Figure 1c suggests that algorithmic improvements may have contributed to the steep improvement in accuracy early on, with increased knowledge of unique structures now being the major determinant. Currently, top-performing algorithms, such as PSIPRED (Jones, 1999), achieve three-state SSP accuracy ∼81%. It has been predicted that SSP will reach an asymptotic accuracy of ∼88% because identical sequence stretches in different tertiary environments can, at times, code for different secondary structures (Kabsch and Sander, 1984; Rost *et al*., 1994; Levin, 1997). Most optimistically, linear extrapolation of Fig. 1a implies that asymptotic accuracies will be achieved in ∼15 years. However, as reasoned later on in this work, the curve may flatten out in the future, further extending the time until asymptotic accuracy will be observed.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
31 JAN 2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Can computationally designed protein sequences improve secondary structure prediction?	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **U.S. Army Medical Research and Materiel Command,Biotechnology High Performance Computing Software Applications Institute,Telemedicine and Advanced Technology Research Center,Fort Detrick,MD,21702**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**7**	

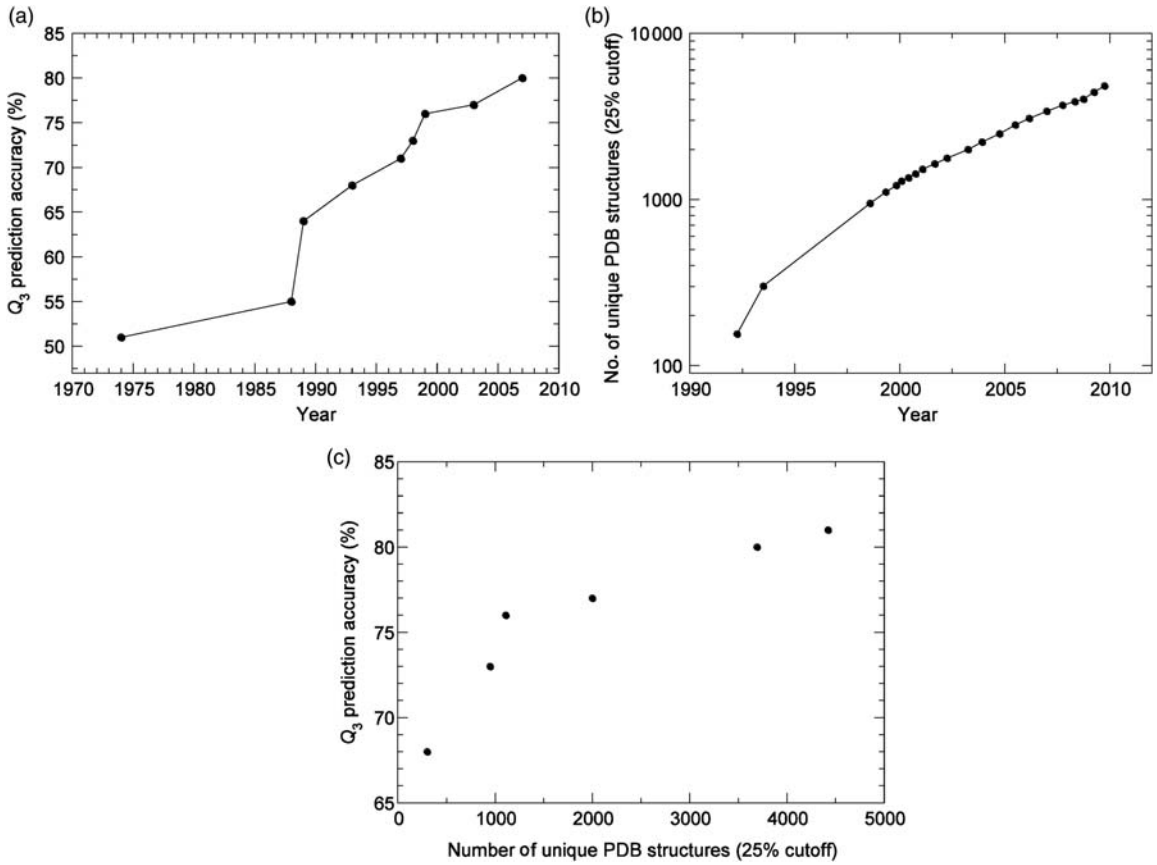

Fig. 1. Historical progression of (**a**) SSP accuracy (Rost, 2003) and (**b**) number of representative PDB structures (25% sequence identity cutoff) (Griep and Hobohm, 2009). (**c**) Synchronistic comparison of prediction accuracy vs. number of representative PDB structures.

Given that it could take ≥15 years to reach the asymptotic accuracy limit, we wondered whether CSD could be used to bolster the database of known sequence–secondary structure correspondences, thereby improving SSP accuracy. More generally, but not addressed in this work, can the space of designed sequences aid in predicting primary protein features such as solvent accessibility, disordered region prediction, sub-cellular localization and other properties that typically use only the sequence or sequence profile as input in the prediction system? Computational protein design provides alternative sequences that are potentially compatible to a given fold. Moreover, a study by Zhang *et al.* (2006) suggested that the known structural (fold) space of single-domain proteins is nearing completeness. Supposing this conjecture is true, there still is the remaining problem of determining the entire space of sequences that could fold up into each unique domain structure. CSD is a plausible means of discovering the sequence space for a given structural domain.

In this work, our hypothesis is that CSD can enrich the database of sequences associated with a given structural domain, thereby significantly enhancing SSP. We use the RosettaDesign program to generate sequences that are compatible with the structural classification of proteins (SCOP) database of known structural domains (Kuhlman and Baker, 2000; Rohl *et al.*, 2004). Secondary structure is predicted using a fuzzy nearest-neighbor algorithm (Bondugula and Xu, 2007). We also investigate whether the designed sequences encode secondary structure correctly in comparison to known protein structures. In the process, we revisit the age-old estimation problem of the theoretical limit of SSP accuracy (Kabsch and Sander, 1984; Rost *et al.*, 1994; Levin, 1997) using a significantly larger database of known structures than previously reported in the literature.

Methods

In this work, the Astral SCOP 1.75 (Murzin *et al.*, 1995; Hubbard *et al.*, 1999) structural domain database filtered at 25% sequence identity was used for both 'training' and testing. In fuzzy nearest-neighbor approaches, there is no training, *per se*, but the database entry matching the query test sequence can be left out. A total of 6511 SCOP 1.75 domains were used after some domains were discarded due to large missing segments ($N_{res} > 10$), non-contiguities in the domain sequence or program failures in any of the design runs. The seven-letter SCOP identifiers used in this work are available as Supplementary data. Secondary structure for each residue in the database was assigned using DSSP (Kabsch and Sander, 1983). The database of native sequences and DSSP-assigned secondary structure is referred to as 'NaturalDB' in the remainder of this work.

The Rosetta suite of programs includes the RosettaDesign module for designing sequences of loops, whole proteins, interfaces, etc., that are compatible with a given structural template (Das and Baker, 2008). The RosettaDesign module consists of an energy function and search components. The energy function, which is an all-atom force field plus an

implicit solvent model that favors hydrophobic amino acids in the core and polar amino acids on the surface, is used to evaluate the suitability of a candidate sequence with a structural template. The search component consists of Monte Carlo optimization with simulated annealing for exploring various amino acid substitutions and side chain conformations. RosettaDesign has been parameterized by the original authors to retain the frequencies of amino acids occurring in the cores and surfaces of naturally occurring proteins. There are two methods available for full-length protein sequence design: the first is to keep the protein backbone fixed and the second allows for some backbone flexibility (Smith and Kortemme, 2008). For each structural domain in the NaturalDB, we generated 50 design sequences using the fixed backbone option and no extra side-chain dihedral sampling (i.e. no *−ex#* options). A sum total of 325 550 (6511 domains × 50 sequences per domain) sequences make up each of our design databases, which we refer to as DesignDBX, where $X = 0$, 35 and 65. DesignDB0 had no restrictions on potential mutations, while DesignDB35 enforced at minimum 35% sequence identity between design and native sequence and, likewise, 65% for DesignDB65. Sequence identity restrictions were imposed by specifying 10 sets of random residue positions not permitted to change. Five sequences were designed for each specification for a total of 50 designs. We did not preferentially choose to keep or create disulfide bridges, thus permitting all 20 amino acids to be substituted at allowed residue positions.

We used a modified fuzzy *k*-nearest-neighbor (FKNN) algorithm described by Bondugula *et al.* (Bondugula and Xu, 2007; Bondugula *et al.*, 2009) to predict secondary structure. The algorithm proceeds in two steps. In the first step, a protein profile is generated by running PSI-BLAST (Altschul *et al.*, 1997) against a large database of protein sequences. Li *et al.* (2002) have shown that sequence identity filtered databases perform better for sequence-property prediction than the complete non-redundant (NR) (Maglott *et al.*, 2000) database. Therefore, we used NR90 to generate a profile of the query sequence. NR90 is a subset of the NR database filtered for sequence identity such that all sequences are <90% identical to each other. The profile is used to search for sequence segments from known protein structures by PSI-BLAST a second time. Depending on the experiment, we used either NaturalDB or DesignDBs in this step. The segments found in the search are used to predict the secondary structure of the input sequence using the modified FKNN algorithm described below.

The prediction, P_i, of residue i of the query sequence is a three-element vector representing the predicted likelihood of the three secondary structure states: helix (H), strand (E) and coil (C). Using the FKNN formula, it is calculated as a weighted average of the actual secondary structure states of the aligned fragments returned from a PSI-BLAST search of the NaturalDB or the DesignDBs, $P_i = \Sigma_{j=1}^{N_{\text{hits}}} B_j^2 \mathbf{S}_{ij} / \Sigma_{j=1}^{N_{\text{hits}}} B_j^2$ where j indexes the hits returned by PSI-BLAST, $\mathbf{S}_{ij}$ is the three-dimensional secondary structure unit vector at position i (i.e. $\mathbf{S}_{ij} = [1\ 0\ 0]$ if the state is 'helix') of the jth fragment hit and B_j is the bit score for the hit j that overlapped with position i. Each fragment is weighted by the square of the alignment bit score returned by the PSI-BLAST program. The exponent of the bit score (which is set to 2) is the only parameter of our 'raw' FKNN approach. Other scores could be potentially incorporated, including expectation value (*E*-value), alignment length and sequence identity. From a computational complexity perspective, the most time-intensive component is the generation of the query PSSM, which is commonly required by most SSP programs. Our FKNN SSP program can be downloaded at http://www.bhsai.org/downloads/fiefpred/.

We benchmarked the approaches with the leave-one-out method of testing, which, in effect, means that the query sequence was no more than 25% identical to any other member of the training set. We report the raw results of the nearest-neighbor algorithm and the filtered results using an artificial neural network (ANN). We also used an ANN to combine various methods. For example, we combined the three-state vectors from our two nearest-neighbor approaches with the output of version 2.4 of the PSIPRED SSP software (Jones, 1999). We trained all ANNs in this work with a small set ($N = 300$) of the 6511 proteins (<5%) leaving 6211 proteins for testing. The ANNs are fully connected feed-forward networks trained using standard error back-propagation algorithms (Haykin, 1998).

We report accuracy results in terms of *Q*-measures, Q_3, Q_{H}, Q_{E} and Q_{C}, defined as follows:

$$Q_3 = 100\% \times \frac{M}{T}$$

$$Q_{\langle \text{H,E,C} \rangle} = 100\% \times \frac{M_{\langle \text{H,E,C} \rangle}}{T_{\langle \text{H,E,C} \rangle}}$$

where M is the number of correctly classified amino acids, T is the total number of amino acids, M_{H} is the number of correctly classified amino acids in the helix configuration and T_{H} is the total number of amino acids in the helix configuration. We investigated the use of other metrics such as segment overlap (Zemla *et al.*, 1999) but found no significant relative differences among methods (results not shown).

As a comparison and an adjunct to the nearest-neighbor algorithm in this work, we chose the popular and top-performing PSIPRED SSP software (Jones, 1999). There are multiple versions of PSIPRED available online as recent as version 3.2. We chose to use PSIPRED v2.4 because it is the latest version for which the training set is also available such that we can clearly differentiate between training and testing sets.

Establishing the upper limit of secondary prediction accuracy using a fragment-based dictionary

We performed a best-case scenario analysis on our secondary structure databases. The question is how accurate would a nearest-neighbor algorithm be if all residue fragments of length N_{res} of a query were found in the database excluding the query sequence itself? In other words, how consistent are the one-to-one mappings of residue strings of length N_{res} to their DSSP-assigned secondary structures (Kabsch and Sander, 1984; Rost *et al.*, 1994). To answer this question, every residue fragment of length N_{res} in the test database was collected as a dictionary of sequence 'words' vs. their corresponding DSSP-derived three-state 'definitions' as seen in

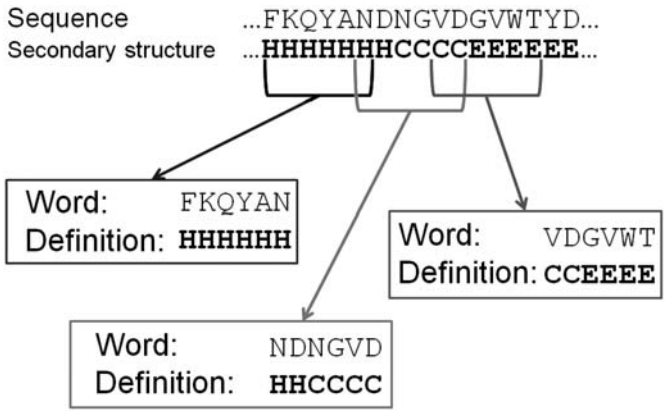

Fig. 2. Schematic for building a dictionary of sequence–secondary structure correspondences of six-residue words.

Fig. 2. Precision for a given word was determined as the ratio of the occurrences of the most populated secondary structure state at a specific position divided by the number of identical words in the database. For example, if every time the word 'AAAERY' translated to the definition 'HHHHHC', then the prediction precision for this word would be 100%. However, if there were four occurrences of 'AAAERY' in the dictionary and one of them differed with the definition 'HHHHHH', then the precision would be 75% for the last residue position of this word and 95.8% overall. In computing the prediction accuracy of words in the DesignDBs, the most common definition in the NaturalDB dictionary, if there was more than one, was considered the correct answer. Finally, due to increased complexity, we did not consider natural sequence variability quantified by substitution matrices (e.g. BLOSUM62) and small alignment gaps.

Results

Design sequence databases, DesignDB0, DesignDB35 and DesignDB65, were generated at imposed minimum sequence identities to native of 0, 35 and 65%. The actual similarity of the sequences to their native counterparts is shown in Fig. 3. The designed sequences in DesignDB0 are, on average, 30% identical to the native sequence, which is in line with previous results of the RosettaDesign developers (Kuhlman and Baker, 2000). DesignDB65, on the other hand, averages 73% identical with a relatively tighter distribution.

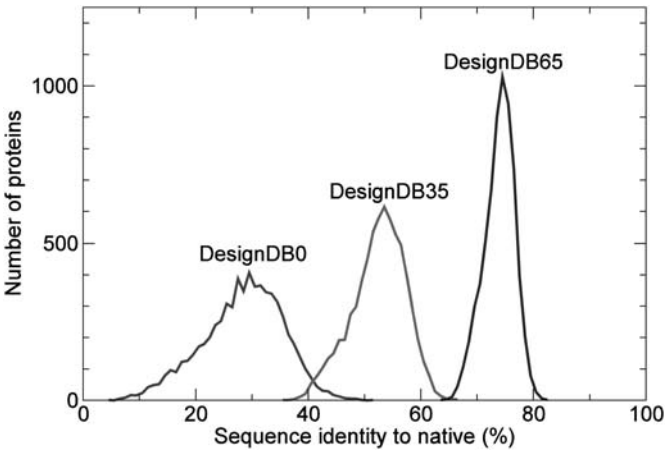

Fig. 3. Distributions of sequence identity of the designed sequences to their natural sequence counterparts averaged over all designs for each protein.

The FKNN algorithm applied to the NaturalDB database provides a reference point for other prediction results. At a raw accuracy of 76.7% (79.7% after ANN filtering), the FKNN algorithm is satisfactory but almost two percentage points inferior to the state-of-the-art, e.g. PSIPRED (Jones, 1999). The primary reason for the inferior results is that the algorithm is quite simple and the neural networks are only trained as a filter of the FKNN output, not on the query sequence or profile. For the FKNN/DesignDB results, the unconstrained DesignDB0 is 1.5% worse than the NaturalDB. This negative result along with the analyses presented below inspired us to constrain the designed sequences by imposing random mutational constraints with a target minimum sequence identity to the native. DesignDB35 yields better results, comparable with the NaturalDB. DesignDB65 shows a 0.4% improvement (one-tailed Wilcoxon signed rank test $P = 2.0144e-90$ at the 0.005 significance level) over NaturalDB. The best results, however, are achieved by combining the NaturalDB and DesignDB predictions, which yield a 1.2% improvement in raw results and a 0.8% enhancement over NaturalDB using an ANN-generated consensus of the two methods.

The simple FKNN algorithm is not expected to perform as well as a mature algorithm such as PSIPRED. Therefore, it is interesting to ask whether an established method can be enhanced by the FKNN algorithm with or without a designed sequence database (Bondugula and Xu, 2007). The consensus of FKNN/NaturalDB and PSIPRED v2.4 is improved 0.7% over PSIPRED alone. The consensus of FKNN/DesignDB65 and PSIPRED v2.4 is even better with a 1.2% enhancement compared with PSIPRED. A one-tailed Wilcoxon signed rank test shows that the neural network consensus of PSIPRED v2.4 and FKNN/DesignDB65 shows a statistically significant ($P = 1.3741e-75$ at the 0.005 level) improvement over the consensus of PSIPRED v2.4 with FKNN/NaturalDB. Intriguingly, the combination of all three methods leads to no net improvement over the two-method consensus. Also, it is worth noting that the strand prediction accuracy, Q_E, of PSIPRED was improved by two percentage points with the inclusion of FKNN/DesignDB65, while the coil prediction accuracy was improved by one percentage point. We compared individual protein predictions from PSIPRED vs. the PSIPRED/DesignDB65 consensus. The consensus method improves upon PSIPRED by reducing or extending strand and helix segments by one to three residues in the direction of closer agreement to the actual observations. This type of variation is consistent with the fact that the ends of predicted secondary structure segments tend to be less reliable (lower scoring) and therefore more apt to state changes when a consensus is made. In contrast, rarely is a new secondary structure segment created or an old segment destroyed. Finally, we made sure that the PSIPRED v2.4 results are not favorably biased by testing on training data. Excluding the 1683 query sequences (of the 6511) that are >25% homologous to any protein in the training set of PSIPRED v2.4, the raw results are virtually identical to those in Table I: $Q_3 = 81.0\%$ for

Table I. Secondary structure prediction accuracies for various protocols. Neural network results are reported for 6211 proteins (301 in training set). Linear combination weights specified in the protocol are only relevant to the Raw results column

Protocol	Raw	Neural network			
		Filter or combination			
	Q_3	Q_3	Q_H	Q_E	Q_C
NaturalDB	76.7	79.7	75	73.5	82.5
DesignDB0	75.1	78.0	73.8	71.4	81.0
DesignDB35	77.1	79.6	75.3	73.9	81.7
DesignDB65	77.7	80.1	74.9	74.2	82.6
$\frac{1}{2}$ (NaturalDB + DesignDB65)	77.9	80.5	75.3	75.0	82.8
PSIPRED v2.4 (Jones, 1999)	81.0	81.4	77.1	75.0	82.4
$\frac{1}{2}$ (NaturalDB + PSIPRED)	81.9	82.1	77.0	76.2	83.2
$\frac{1}{2}$ (DesignDB65 + PSIPRED)	82.5	82.6	77.4	77.0	83.5
$\frac{1}{3}$ (NaturalDB + DesignDB65 + PSIPRED)	82.3	82.5	77.2	77.1	83.5

PSIPRED alone and $Q_3 = 82.4\%$ for the equally weighted combination of PSIPRED v2.4 and FKNN/DesignDB65.

Looking at the distribution of alignment lengths and bit scores from 100 protein queries in Fig. 4a and b, we see that the design database improves both the alignment length and the bit scores of the top 2000 hits for a given query. As impressive as this appears, the increased hit quality is most likely a function of the fact that the DesignDBs are 50 times larger than the NaturalDB. Our conjecture can be surmised by the fact that the distributions of the DesignDBs are identical even though DesignDB65 is clearly more constrained in sequence space than the others (as seen in the Dictionary analysis section).

Dictionary analysis

Given that the DesignDBs provide significantly more search space compared with NaturalDB, why did prediction accuracy not improve more significantly? To answer this question, we analyzed the dictionaries of sequence-property correspondences (Fig. 2). First, with our large 6511 SCOP domain data set, we revisit the question 'how self-consistent are the three-state definitions of the native sequences?' Tabulating all exact matches of sequence words of a given length, what percentage of corresponding three-state secondary structure strings matched the consensus? As can be seen in Fig. 5a, the precision of exact string matches tops out ~90%, which is close to the prediction (~88%) made more than a decade ago (Kabsch and Sander, 1984; Rost *et al.*, 1994). Furthermore, in Fig. 5c, it can be seen that the unique word count as a percentage of total possible words drops precipitously after five-residue strings (i.e. pentapeptides). This may explain why we reach only ~80% Q_3 accuracy using the NaturalDB. This result also does not bode well for future improvements in prediction accuracy because large increases in sequence space may be required to boost recognition of longer words and correspondingly more precise secondary structure definitions.

In contrast, the unconstrained design set, DesignDB0, tops out at an 'asymptotic' accuracy of 82.7% with 10-residue words (Fig. 5a), in accordance with its inferior predictive accuracy. Moreover, even if 90% asymptotic accuracy could be obtained by this DesignDB, the word space of only one additional residue is nearly complete ($N_{res} = 6$) as seen in Fig. 5c, where the number of unique words as a fraction of all possible words appears to be shifted by about a residue. As alluded to before, correctly assigned 10-residue words are required to reach the pinnacle of 90% asymptotic accuracy, which would presumably require a significantly larger DesignDB ($\sim 20^4 = 160\,000\times$ larger, at least). Sequence space limitations notwithstanding, we reasoned that placing constraints on mutations would increase the asymptotic accuracy achieved by a designed sequence database. By definition, in the limit of 100% sequence identity constraints (i.e. no mutations), the 90% accuracy limit would be obtained as DesignDB becomes equivalent to NaturalDB. As expected, imposed sequence identities of 35 and 65% lead to improved asymptotic accuracies at the 10-residue word length (83.8 and 84.5%, respectively). Unfortunately, as the mutational constraints are increased, the diversity of the designed sequences drops. For reference, the unconstrained DesignDB0 has ~40 times more words than NaturalDB (Fig. 5b), which translates to a little more than a single amino acid increase in dictionary coverage based on Fig. 5c. In comparison, as seen in Fig. 5b, DesignDB35 has on average 10% fewer unique words than the unconstrained

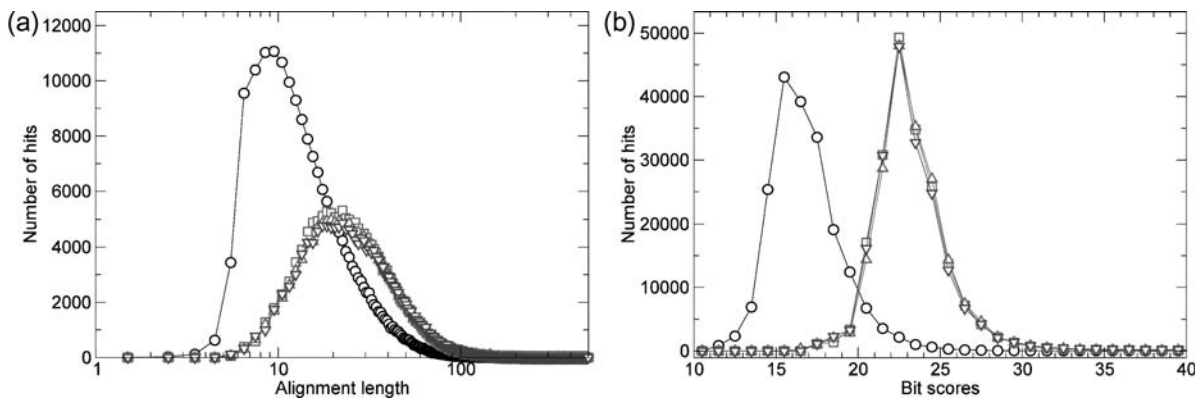

Fig. 4. Distributions of (**a**) alignment lengths and (**b**) bit scores for 100 PSI-BLAST queries of the NaturalDB (circles) and DesignDBs (excluding the hits from the query sequence and query designs). Legend: squares—DesignDB0, up triangles—DesignDB35, down triangles—DesignDB65.

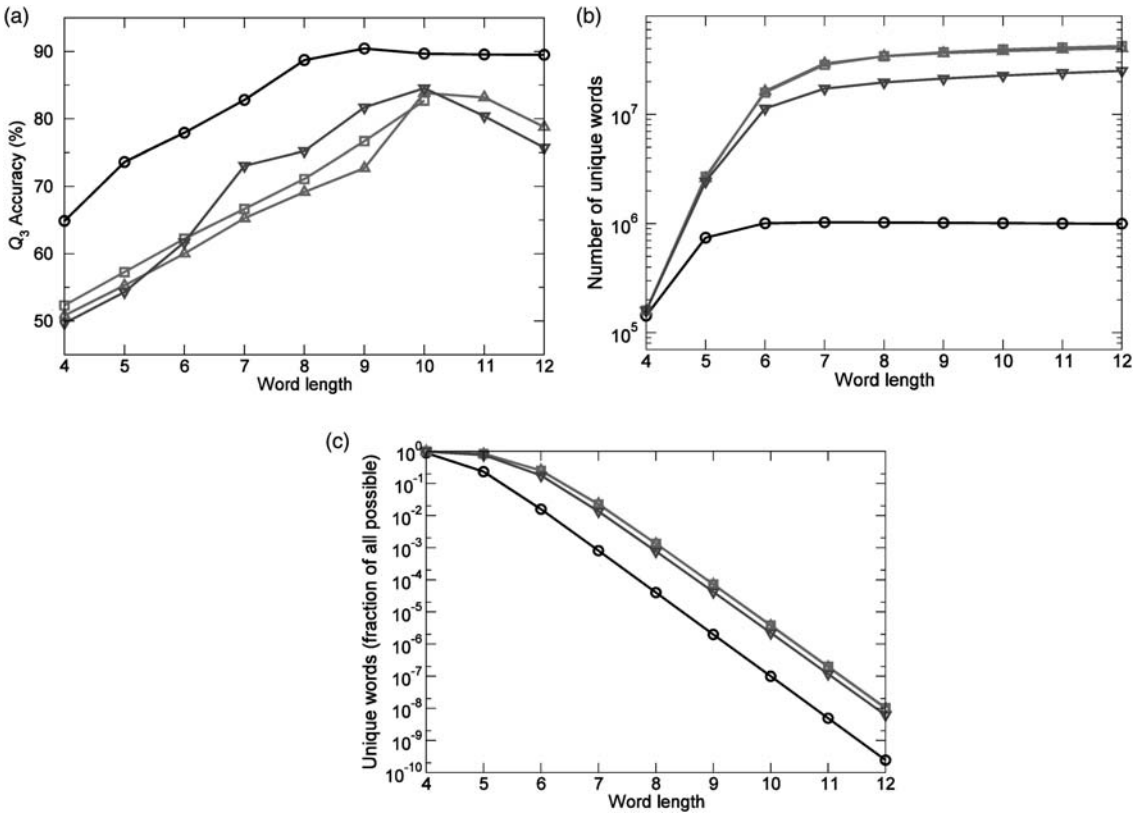

Fig. 5. Dictionary analysis of the sequence–secondary structure databases: (**a**) accuracy of exact sequence matches compared with consensus secondary structure of native sequences as a function of word length, (**b**) number of unique words of a given length, and (**c**) unique words as a fraction of all possible words in the databases for each length. In the native sequence analysis, the query sequence is not included in the consensus. In the design sequence analysis, matches with parent native sequence are not included in the statistics. Legend: circles—NaturalDB, squares—DesignDB0, up triangles—DesignDB35, down triangles—DesignDB65.

DesignDB0 at various word lengths; DesignDB65 has ~40% fewer unique words than DesignDB0. All in all, there is a trade-off between sequence diversity and asymptotic accuracy for the different DesignDBs.

Discussion

The hypothesis of this work was that a set of designed sequences for each structural domain in a database would increase the number of nearest-neighbor matches to a query sequence, thereby enhancing the prediction of secondary structure. Our experiments indicate that the enhancement associated with the development of a designed sequence database is statistically significant, but well below the theoretical limits of prediction accuracy. Furthermore, while the 1% overall improvement on PSIPRED is appealing, consensus predictions with other methods in the literature may yield similar gains.

Why weren't prediction accuracies improved more? Designed sequences with no functional constraints tend to deviate from the natural space of sequences (Cheng *et al.*, 2005) and overemphasize the importance of structural stability. Moreover, the loss of functionally conserved residues upon backbone redesign will reduce the 'realness' of unconstrained computationally designed sequences. Also, it has been previously noted that computationally designed sequences are somewhat different from native sequences. In an early study of RosettaDesign, the average sequence identity between the design and native sequences in the core region of a protein was ~55%, while the overall sequence identity was ~35% (Liu and Kuhlman, 2006) in accord with our results. Nature 'designs' proteins based on a variety of factors, including thermal stability and function. For example, it has also been observed that computationally designed proteins are often more stable than the native proteins from which their original backbone templates were derived (Dantas *et al.*, 2003). This can partially be rationalized by the fact that extremely stable proteins are unfavorable for organisms which must regularly turn over many of its proteins through cellular recycling.

In addition, design programs do not have the perfect scoring function for thermodynamic stability. Even with an optimal potential energy function, a given design model has to be evaluated based on its free energy of stability in the basin of the desired backbone conformation. To compute this, molecular dynamics or Monte Carlo simulations would be needed, but these methods are orders of magnitude slower than the actual design algorithm, and thus currently unfeasible for large-scale generation of designed sequences.

The results for nearest-neighbor matching to a database of designed sequences were quite good, considering that designed sequences tend to drift away from natural sequences (Kuhlman and Baker, 2000). Because designed sequences offer a range of possible sequences for a given backbone template, they can be especially useful when a protein has few known sequence homologues. We could not find a

scheme better than random selection to cull out the 'best' designed sequences, such as filtering with an alternative scoring function (Lee and Olson, 2007) or similarity to native. Therefore, we simply used the whole collection in our search. We also did not choose to fix the small clusters of residues that might be important for function because this seemed irrelevant for SSP. Finally, we tried using 200 designed sequences per protein domain (instead of 50), but found that the results from the FKNN algorithm did not materially improve (results not shown). Most likely, the extra sequences did not significantly increase diversity because of the fixed backbone limitation. Therefore, it may be worthwhile to investigate the use of larger sets of designed sequences while incorporating backbone flexibility.

In order to understand why prediction accuracy enhancement due to computational design was not more pronounced, we performed a simple analysis by building a dictionary of correspondences between residue strings and secondary structure state strings. Using this approach, we found that unconstrained designed sequences (DesignDB0) increased the number of unique words by up to a factor of 40, but sometimes associated with the wrong secondary structure strings, thus reducing the asymptotic accuracy of the design sequence database from 90 to 82.7%. By imposing constraints on the fraction of allowable mutations, we were able to recover ~2% of the asymptotic accuracy without significantly lowering the number of unique words. The best designed sequence database, DesignDB65 yielded a modest 0.8% accuracy improvement compared with only using natural sequences. Most likely, the mutation constraints compensate for limitations of the sequence design objective function in RosettaDesign. Computational design in tandem with backbone optimization has been argued to be a more accurate approach (Larson *et al*., 2002). However, due to the greatly increased computational complexity of this approach, we did not pursue it in this pilot work.

In conclusion, sequences designed from a large set of fixed backbone protein domains modestly enhance SSP accuracy. Furthermore, the combination of PSIPRED and FKNN/NaturalDB led to a small improvement over PSIPRED alone. There appear to be two primary limitations to our current approach. First, unconstrained computational design leads to errors in predicted secondary structure compared with a natural database benchmark. This can be rectified to some extent by introducing constraints in the percentage of residues that can be mutated. Second, the sequence space necessary to achieve an accurate and complete 10-residue word lookup table of secondary structure is still several orders of magnitude larger than the computationally designed sequence databases used in this work.

Supplementary data

Supplementary data are available at *PEDS* online.

Acknowledgements

We would like to thank Dr Daniel Ripoll for helpful discussions.

Conflict of interest: The opinions and assertions contained herein are the private views of the authors and are not to be construed as official or as reflecting the views of the US Army or of the US Department of Defense. This paper has been approved for public release with unlimited distribution.

Funding

This work was sponsored by the US Department of Defense High Performance Computing Modernization Program, under the High Performance Computing Software Applications Institutes Initiative.

References

Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) *Nucleic Acids Res.*, **25**, 3389–3402.
am Busch,M.S., Mignon,D. and Simonson,T. (2009) *Proteins*, **77**, 139–158.
Berman,H.M., Battistuz,T., Bhat,T.N., *et al*. (2002) *Acta Crystallogr. D Biol. Crystallogr.*, **58**(Pt 6), 899–907.
Bondugula,R. and Xu,D. (2007) *Proteins*, **66**, 664–670.
Bondugula,R., Lee,M.S. and Wallqvist,A. (2009) *Nucleic Acids Res.*, **37**, 452–462.
Cheng,G., Qian,B., Samudrala,R. and Baker,D. (2005) *Nucleic Acids Res.*, **33**, 5861–5867.
Chou,P.Y. and Fasman,G.D. (1974) *Biochemistry*, **13**, 222–245.
Dantas,G., Kuhlman,B., Callender,D., Wong,M. and Baker,D. (2003) *J. Mol. Biol.*, **332**, 449–460.
Das,R. and Baker,D. (2008) *Annu. Rev. Biochem.*, **77**, 363–382.
Griep,S. and Hobohm,U. (2010) *Nucleic Acids Res.*, Epub September 25, 2009.
Haykin,S. (1998) *Neural Networks: A Comprehensive Foundation*. Prentice Hall,Upper Saddle River, NJ.
Hubbard,T.J., Ailey,B., Brenner,S.E., Murzin,A.G. and Chothia,C. (1999) *Nucleic Acids Res.*, **27**, 254–256.
Jones,D.T. (1999) *J. Mol. Biol.*, **292**, 195–202.
Kabsch,W. and Sander,C. (1983) *Biopolymers*, **22**, 2577–2637.
Kabsch,W. and Sander,C. (1984) *Proc. Natl Acad. Sci. USA*, **81**, 1075–1078.
Koehl,P. and Levitt,M. (2002) *Proc. Natl. Acad. Sci. USA*, **99**, 691–696.
Kuhlman,B. and Baker,D. (2000) *Proc. Natl. Acad. Sci. USA*, **97**, 10383–10388.
Larson,S.M., England,J.L., Desjarlais,J.R. and Pande,V.S. (2002) *Protein Sci.*, **11**, 2804–2813.
Larson,S.M., Garg,A., Desjarlais,J.R. and Pande,V.S. (2003) *Proteins*, **51**, 390–396.
Lee,M.S. and Olson,M.A. (2007) *J. Chem. Theory Comput.*, **3**, 312–324.
Levin,J.M. (1997) *Protein Eng.*, **10**, 771–776.
Li,W., Jaroszewski,L. and Godzik,A. (2002) *Protein Eng.*, **15**, 643–649.
Liu,Y. and Kuhlman,B. (2006) *Nucleic Acids Res.*, **34**, W235–W238.
Maglott,D.R., Katz,K.S., Sicotte,H. and Pruitt,K.D. (2000) *Nucleic Acids Res.*, **28**, 126–128.
Murzin,A.G., Brenner,S.E., Hubbard,T. and Chothia,C. (1995) *J. Mol. Biol.*, **247**, 536–540.
Pei,J., Dokholyan,N.V., Shakhnovich,E.I. and Grishin,N.V. (2003) *Proc. Natl. Acad. Sci. USA*, **100**, 11361–11366.
Rohl,C.A., Strauss,C.E., Misura,K.M. and Baker,D. (2004) *Methods Enzymol.*, **383**, 66–93.
Rost,,B. (2003) Rising accuracy of protein secondary structure prediction. In Chasman,D. (ed.), *Protein Structure Determination, Analysis, and Modeling for Drug Discovery*. Dekker, New York; pp. 207–249.
Rost,B., Sander,C. and Schneider,R. (1994) *J. Mol. Biol.*, **235**, 13–26.
Schmidt Am Busch,M., Sedano,A. and Simonson,T. (2010) *PLoS One*, **5**, e10410.
Smith,C.A. and Kortemme,T. (2008) *J. Mol. Biol.*, **380**, 742–756.
Zemla,A., Venclovas,C., Fidelis,K. and Rost,B. (1999) *Proteins*, **34**, 220–223.
Zhang,Y., Hubner,I.A., Arakaki,A.K., Shakhnovich,E. and Skolnick,J. (2006) *Proc. Natl. Acad. Sci. USA*, **103**, 2605–2610.